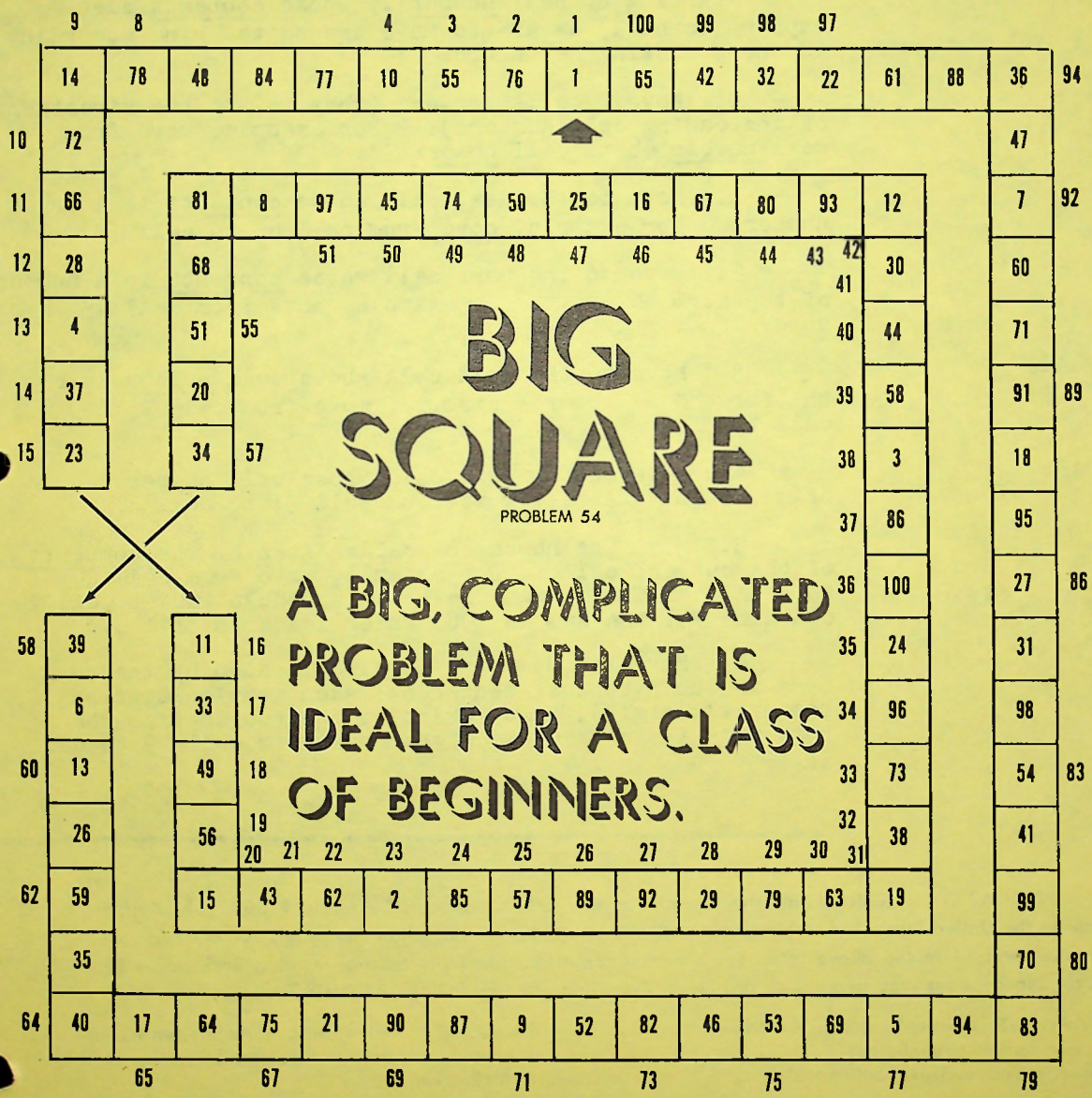
BOX 272
CALABASAS, CA
91302

Popular Computing

VOL 2 NO 8
AUGUST 1974



BIG SQUARE

The diagram shown on the cover is essentially a closed ring of 100 cells. The cells are numbered counterclockwise, starting at the arrow. Within the cells is the set of numbers from 1 to 100 in scrambled order.

Starting at cell number 1, whose contents also happens to be 1, we are to move around the ring according to the following set of rules:

——→ 0. Move 7 cells forward (that is, in the direction of increasing cell numbers). For example, move from cell number 68 to cell number 75.

——→ 1. Move to the next cell whose contents is a prime number. For example, move from cell 65 to cell 75.

——— 2. Move to the next cell whose contents is a number of the form 8K - 1. For example, move from cell 45 to cell 58.

——— 3. Move to the next cell whose cell number is of the form 8K - 1. For example, move from cell 15 to cell 23.

——→ 4. Move to the next cell whose cell number is prime. For example, move from cell 31 to cell 37.

——— 5. Move the number of cells given by the contents of the current cell. For example, move from cell 29 to cell 8. The arithmetic here is modulo 100. Cell 29 contains 79; the sum is 108, which leads to cell 8.

——→ 6. Move the number of cells dictated by the next term in the Fibonacci sequence (taken modulo 10; that is, the unit's digit). For example, the fourth time that this rule is invoked, we might move from cell 29 to cell 37.

POPULAR COMPUTING is published monthly at Box 272, Calabasas, California 91302. Subscription rate in the United States is $18 per year, or $15 if remittance accompanies the order. For Canada and Mexico, add $4 to the above rates. For all other countries, add $6 to the above rates. Back issues $2 each. Subscriptions may begin with any issue.

Publisher: Fred Gruenberger
Editor: Audrey Gruenberger
Associate editor: David Babcock

Contributing editors: Richard Andree, Irwin Greenwald, Daniel D. McCracken, William C. McGee

Advertising manager: Ken W. Sims
Art director: John G. Scott

——→ 7. Move to the next cell whose contents is of the form 8K + 3. For example, move from cell 89 to cell 16.

—— 8. Move back to the next odd-numbered cell. For example, move from cell 31 to cell 29.

——→ 9. Same as rule 6. For either rules 6 or 9, if the next digit in the Fibonacci sequence is zero, apply rule zero (i.e., move ahead 7 cells) and arrange to proceed to rule 1 next.

These ten rules are to be applied in rotation, starting with rule zero.

Here are the problems:

A. At what cell will we be after the 1000th move?

B. The rules are taken in rotation, and the Fibonacci sequence repeats, as explained below. Eventually, the pattern of moves must repeat. What is the cycle length of this repetition?

C. What is the distribution of the numbers of the cells that are visited? (In the first 200 moves, 36 of the cells are not visited, and cell 85 is visited eleven times.)

The three Big Square problems make an ideal project for an introductory class. Each of the ten move rules can be written as a subroutine, and these subroutines can be (and should be) coded, debugged, and tested independently. Another subroutine is needed, to generate the Fibonacci sequence modulo 10.

These subroutines can interact. Two of them call for testing a number between 1 and 100 for primality. Two of them call for testing a number between 1 and 100 to determine whether or not it is of the form 8K - 1. In both these cases, besides the common logic, there is a question of judgement. To determine primality, for example, is it better to divide by 2, 3, 5, and 7, or to look up a table of the 25 prime numbers less than 100? And better in what sense?: ease of coding; ease of checkout; or machine efficiency?

The unit's digit of the Fibonacci sequence repeats on a cycle of 60. Again, is it better to generate the sequence, modulo 10, or store the 60 elements and look them up as needed?

The Problem is designed to teach these concepts:

1. Problem segmentation through subroutining; systems testing of the entire program after each module has been tested.

2. Construction of careful and thorough test procedures for each module.

3. The distinction between an address and the contents at that address; the distinction between where it is and what it is.

4. To show that a simple, well-defined procedure can be carried out to any extent readily and cheaply by computer, where the same procedure carried out by hand would be tedious, error-prone, or even impossible.

The first 30 moves are given below as test data.

Move rule	Cell No.	Contents	Cell No.	Contents	Cell No.	Contents
0	8	78	25	57	44	80
1	14	37	26	89	45	67
2	15	23	29	79	58	39
3	23	2	31	19	63	35
4	29	79	37	86	67	75
5	8	78	23	2	42	12
6	10	72	28	29	45	67
7	16	11	31	19	55	51
8	15	23	29	79	53	81
9	18	49	37	86	54	68

In addition, the positions after certain numbers of moves are given here as test data. Note that in this problem, 1 is not considered a prime number:

Move number	Cell number	Contents
50	19	56
100	23	2
150	55	51
200	86	27
500	31	19
800	20	15

Speaking of Languages

ROBERT TEAGUE

The column this month is intended to provoke some thought, anticipation, and some help for a forthcoming Fortran exercise. There is much discussion over what is and what is not Fortran. Perhaps the better question is: How close to the standard does the compiler come? Certainly there are some areas where all compilers will be up to snuff. There are other areas, however, where many compilers fail to live up to the standard. The examples given here would be nitpicking in their details, perhaps, but do indicate whether the compiler can be trusted to handle all situations properly or not. If a compiler doesn't handle a difficult situation, it will cast some suspicion as to whether or not it will handle the easier situations. After all, "whatever can go wrong, will..."

These examples are given to help precipitate some thought along the lines discussed above, but also to elicit some help. Many times we develop a pet statement which tricks or confuses a compiler and is our personal test of how good a compiler is. If any reader has one or more of these, I would appreciate seeing them. In a future issue, a test program to give a Fortran compiler a thorough workout will be presented, and these volunteered statements could be a big boon. Send all correspondence to: Speaking of Languages, Popular Computing, Box 272, Calabasas, California 91302.

This first Fortran statement was submitted by Allen Brady of the University of Nevada at Reno:

```
110  FØRMAT (X3H) = (A4)
```

If FØRMAT is a dimensioned variable, this should compile as an assignment statement. The variable X3H is used as the subscript, while the expression A4 is enclosed in an unnecessary (but not incorrect) set of parentheses.

Consider the following as well:

```
      DIMENSIØN WRITE (10,10)
      WRITE (6,2) = X
      WRITE (6,2)
    2 FØRMAT (9H IT WØRKS)
```

The context should allow the compiler to handle both situations correctly. The first, because of the equals sign, should be interpreted as a reference to the subscripted variable in an assignment statement. Also due to its syntactical usage, the second statement should be taken as an actual WRITE statement.

Although we will be primarily concerned with Fortran in the forthcoming column, this same variance of compiler from standard occurs in most languages. Consider the following in CØBØL:

```
77 K PICTURE 9 USAGE CØMPUTATIØNAL.

PERFORM PP-2 VARYING K FROM 1 BY 1 UNTIL K = 10.
```

As shown in an earlier issue (PC9-13), this should produce an infinite loop, since K can't take on a value of 10 in only a single digit. But with the data-name K being computational, many compilers (especially on a fixed-word binary machine) will handle the loop properly because K is stored in more than a single digit (usually a whole word).

There are many other such situations, but the ones cited should stimulate thinking along the right path.

N-SERIES 17

Log 17 1.230448921378273928540169894328337030007567378425046

Ln 17 2.833213344056216080249534617873126535588203012585745

$\sqrt{17}$ 4.123105625617660549821409855974077025147199225373620

$\sqrt[3]{17}$ 2.571281590658235355453187208739726116427901632454696

$\sqrt[5]{17}$ 1.762340347832317013861002253564869928083029281958161

$\sqrt[7]{17}$ 1.498919872071562012172790123682529677859722797940226

$\sqrt[10]{17}$ 1.327531674888519190643256886021207526009763060794685

$\sqrt[100]{17}$ 1.028737305747143922900350475730879422626305236635283

e^{17} 24154952.753575298214775435180385823879867567352722517379751434218432326857035873328770784467537 3

π^{17} 282844563.586533053154230561383867746094227366505669287617018839241290062307703576929778692851 46

$\tan^{-1}$ 17 1.512040504079173926329138389187979656621934281587108264343969733186463982719911941311562388925826973252

17^{100} 1108899372780783641306111715875094966436017167649879524402769841278887580501366697712424694256005093589248451503068397608001

The IBM Card Problem

NUMBER 55

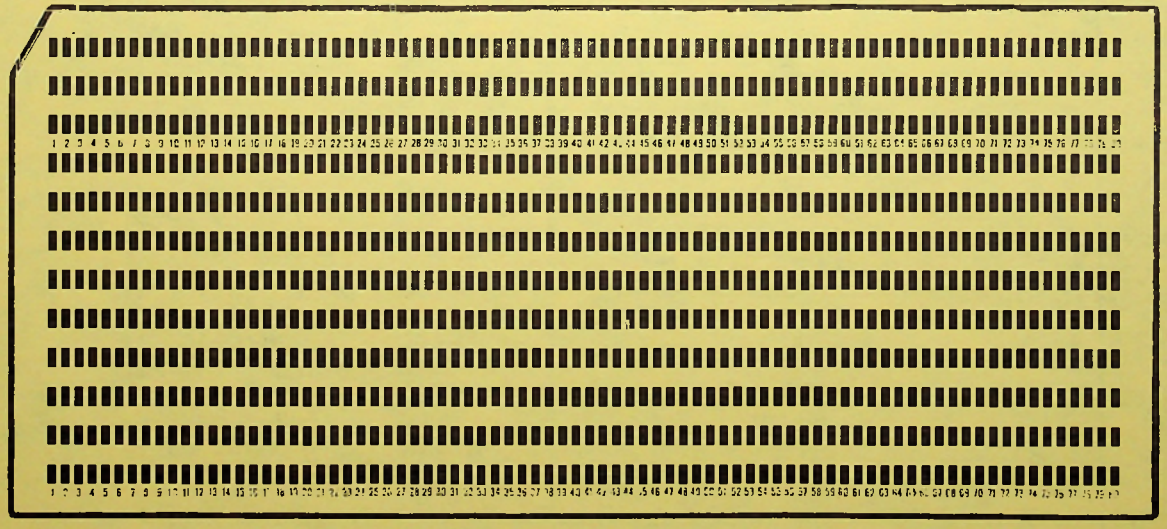
The holes in an IBM card form a 12 x 80 grid. The rows are identified 12, 11, 0, 1, 2, ... , 9 and the columns are numbered from 1 to 80. Expressing the coordinates of a given hole in the usual (X,Y) fashion, the hole at the lower left corner is (1,9) and the hole in the upper right corner is (80,12).

Moving always from left to right (that is, from column 1 toward column 80) and from bottom to top (that is, from the 9 row toward the 12 row), in how many ways may one move from (1,9) to any other hole? The number of ways to go from (1,9) to any hole within the square up to (12,12) is given by Pascal's triangle. Thus, to go from (1,9) to (11,5), there are 1001 possible paths. From (1,9) to (12,12) there are 705432 possible paths. From (1,9) to (40,12) there are over 37 billion paths.

(A) What is the logic of a subroutine that furnishes the number of paths from (1,9) to (column X, row Y)?

(B) Write a Fortran program for that logic.

(C) What is the number of paths from (1,9) to the hole at (80,12)?



Problem Solution

Gauss's Lattice Problem (Number 48, Issue No. 14) requires determining the number of lattice points (Q) in or on a circle of radius R centered at the origin.

David E. Ferguson, President of Group/3 (a division of Informatics Inc.) developed the following closed form expression for Q as a function of R:

$$\text{Let } k = \left[R/\sqrt{2}\right]$$

$$Q = 1 + 4(R-k) + 4k(k+1) + 8\sum_{n=1}^{R-k-1}\left[\sqrt{n(2R-n)}\right]$$

The formula holds true for all integral values of R. If R is replaced by R in the first two occurrences of R in the formula, then the formula holds true for all real R. For example, for R = 4,

$$k = 2$$

$$Q = 1 + 4(4-2) + 4\cdot 2(3) + 8\left[\sqrt{1(8-1)}\right]$$

= 49, as can be readily verified by observation.

Mr. Ferguson programmed the formula to run on his System/3 (a machine which does not have multiply or divide op-codes), and obtained the following results in 200 seconds of CPU time:

5	81	4000	50265329
6	113	5000	78539677
7	149	6000	113097185
8	197	7000	153937805
9	253	8000	201061681
10	317	9000	254468477
20	1257	10000	314159053
30	2821	20000	1256636857
40	5025	30000	2827432965
50	7845	40000	5026547529
60	11289	50000	7853981045
70	15373	60000	11309732881
80	20081	70000	15393802989
90	25445	80000	20106192121
100	31417	90000	25446899381
200	125629	100000	31415925457
300	282697	200000	125663704421
400	502625	300000	282743337729
500	785349	400000	502654823345
600	1130913	500000	785398158957
700	1539297	600000	1130973352337
800	2010573	700000	1539380397873
900	2544569	800000	2010619296341
1000	3141549	900000	2544690045473
2000	12566345	1000000	3141592649625
3000	28274197		

Problem Solution

The following problem appeared in issue No. 8, page 16: Determine two rational fractions the sum of whose cubes is 6. The solution below is by David Ferguson, President of Group/3.

If $(a/b)^3 + (c/d)^3 = 6$ it can be assumed that $(a,b) = (c,d) = 1$. Then

$$a^3d^3 + c^3b^3 = 6b^3d^3$$

from which it follows that $d|b$ and $b|d$, so $b = d$. Therefore the problem can be restated as

$$a^3 + c^3 = 6b^3$$

where a, b, and c are integers for which $(a,b) = (c,b) = 1$.

But any common factor of a and c must divide b, so that a and c are relatively prime, and hence they are both odd. Any two relatively prime odd integers can be written as:

$$a = p + q \qquad c = p - q \qquad (p,q) = 1.$$

Then

$$6b^3 = (p + q)^3 + (p - q)^3$$

$$6b^3 = 2p(p^2 + 3q^2)$$

$$3|p \quad = 3r$$

$$b^3 = 3r(q^2 + 3r^2)$$

Since $3|p$ and $3\nmid q$

$$3|b = 3e, \quad 3^2|r = 3^2s$$

Then $e^3 = s(q^2 + 3^5s^2)$ since $s|p$, $(s,q) = 1$.

So $s = t^3$, $e = ft$, and $f^3 = q^2 + 3^5t^6$.

The smallest numerical substitution, $t = 1$, yields a solution

$$f^3 = q^2 + 243.$$

The smallest cube greater than 243 is $7^3 = 343$, which yields $q = 10$; then

$p = 3^3t^3 = 27$; $b = 3ft = 21$; $a = p+q = 37$ and $c = p-q = 17$.

Mr. Ferguson has solved the problem analytically, thus spoiling all the fun. A computer solution by Mary Bruskotter is more appealing to computists:

From the deduction, as before, that

$$a^3 + b^3 = 6c^3$$

(for the fractions a/c and b/c), start with a = 1 and b = 2c - 1 for a given value of c. Evaluate both sides of equation (A), and proceed with this algorithm:

LEFT > RIGHT: b - 1 replaces b;
LEFT < RIGHT: a + 1 replaces a.

When a becomes greater than b, c + 1 replaces c. The solution will be obtained when LEFT = RIGHT. This first occurs when a = 17, b = 37, and c = 21 (or a = 119, b = 259, and c = 147, and so on for infinitely many solutions).

A New Irrational PROBLEM 56

List the decimal proper fractions in lowest terms in order by denominators, and within the same denominator, in order by numerators. (The array was shown in PC3-9 as the "Number the Fractions Problem")

Convert each of these fractions to binary. Form a new binary fraction by taking the first bit from the first fraction, the second bit from the second fraction, and so on. The derivation of the new fraction is shown by the circled bits in the diagram.

What is the decimal value of the resulting number? (It is approximately .88900...; the Problem here is to determine its value correct to 20 decimal places.)

1/2	.(1)0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1/3	. 0(1)0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
2/3	. 1 0(1)0 1 0 1 0 1 0 1 0 1 0 1 0 1
1/4	. 0 1 0(0)0 0 0 0 0 0 0 0 0 0 0 0 0
3/4	. 1 1 0 0(0)0 0 0 0 0 0 0 0 0 0 0 0
1/5	. 0 0 1 1 0(0)1 1 0 0 1 1 0 0 1
2/5	. 0 1 1 0 0 1(1)0 0 1 1 0 0 1 1
3/5	. 1 0 0 1 1 0 0(1)1 0 0 1 1 0 0
4/5	. 1 1 0 0 1 1 0 0(1)1 0 0 1 1 0
1/6	. 0 0 1 0 1 0 1 0 1(0)1 0 1 0 1
5/6	. 1 1 0 1 0 1 0 1 0 1(0)1 0 1 0
1/7	. 0 0 1 0 0 1 0 0 1 0 0(1)0 0 1
2/7	. 0 1 0 0 1 0 0 1 0 0 1 0(0)1 0
3/7	. 0 1 1 0 1 1 0 1 1 0 1 1 0(1)1

- Shortcut:

Since many desk and pocket calculators have a square root function, it is expedient to capitalize on that function in the calculation of cube roots. The familiar Newton derivation for cube root proceeds from

$$x^3 - N = 0$$

and produces the recursion

$$x_{n+1} = \frac{2x_n^3 + N}{3x_n^2}$$

If, however, the derivation starts with

$$x^{3/2} - N^{1/2} = 0$$

we then have

$$x_{n+1} = \frac{x_n}{3} + \frac{2\sqrt{N}}{3\sqrt{x_n}}$$

or

$$x_{n+1} = \frac{x_n}{3} + \frac{2\sqrt{N}\sqrt{x_n}}{3x_n}$$

For the cube root of 2, these two approaches show the following calculations:

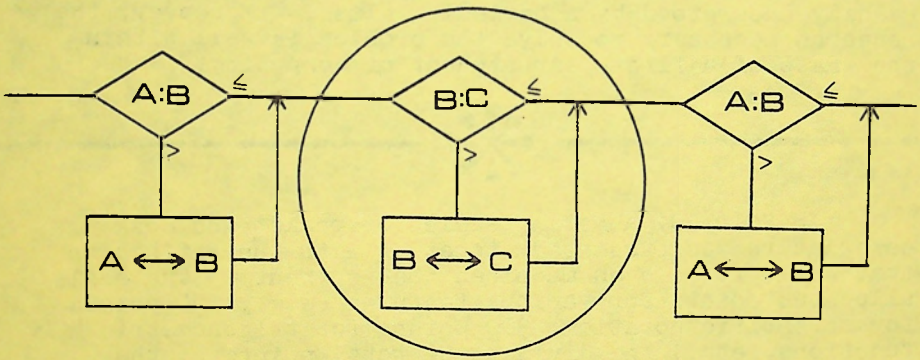
	Usual formula	Square root formula
x_0	1	1
x_1	1.33333	1.276142
x_2	1.26388	1.2599729
x_3	1.2599329	1.2599208

(true value = 1.25992104989487...)

Thus, by utilizing the available square root function, the convergence to cube root can be speeded up.

A Problem of Another Sort NUMBER 57

To sort three numbers, A, B, and C, by direct internal sorting, the scheme shown in Figure A will do. There are other possibilities for the comparisons and interchanges that have to be made (for example, the comparisons A:B, A:C, and B:C will also work), but there must be three of them.



A

For convenient reference, let us adopt the notation BC to denote the logic within the large circle in Figure A.

To sort four numbers, A, B, C, and D, by a scheme analogous to that of Figure A, we would use:

AB BC CD AB BC AB

This will work and, indeed, many books have indicated that six comparisons are necessary. But the following set:

AB CD AC BD BC

are sufficient to perform the sort.

Consider then the problem of direct internal sorting of five numbers, A, B, C, D, and E. We can immediately devise a scheme that works:

AB BC CD DE AB BC CD AB BC AB

but the previous work on the four-word problem suggests that less than ten comparisons will suffice. It is not difficult to find a set of nine:

AB BC CD DE AB CD AC BD BC

(The first four insure that the largest number has been moved all the way to the right; the other five are simply the scheme for four things repeated again.) A similar scheme is the following:

DE CD BC AB BC DE BD CE CD

In summary, to sort five things, the number of comparisons and possible interchanges is certainly greater than six, but may be less than nine. The minimum number is not presently known, nor is a method of finding it (other than massive brute force, involving astronomical trials).

There is a problem here, and it is one that can be readily understood by a beginning class. Moreover, the research necessary to solve the problem is well within the grasp of a first-year student of computing.

In PC10-10, a rating scale for pocket and desk calculators was presented, together with the ratings on that scale for eleven machines then current. The scale allocates points for various features (e.g., 200 points for scientific notation; 300 points for trigonometric functions, etc.) and divides the total points by the price of the machine. Thus, as the prices drop, the ratings go up.

And the prices have dropped. Competition has become keen, to the point where several manufacturers are presenting lists of features of comparable machines, with their competitors' machines identified by model number. The table on the facing page shows the range of ratings for the most widely sold machines. The basic unit is what the industry calls a "Four-banger," which has the four arithmetic functions. Most such machines now have floating point and 8 digits for their calculations and display. At $30 (e.g., the Litronix 1100) the rating would be 11.67, which is higher than any machine listed only six months ago.

Ratings of some current machines (at their present prices) are given here

Bowmar MX100	9.38
TI SR-10	12.00
HP-80	8.94
HP-35	6.58
TI SR-50	14.94
Execucal M55	7.00
Kingspoint Scientific	7.78
Sperry Remington SSR-8	11.78
Electronic Slideruler	16.00
Sears 5885	7.16
Bowmar MX55	7.64

Price ———	30	40	50	60	70	80	90	110	120
Basic Four-banger	11.67	8.75	7.00	5.83	5.00	4.38	3.89	3.18	2.92
4B + reciprocals	13.33	10.00	8.00	6.67	5.71	5.00	4.44	3.64	3.33
4B + square root	15.00	11.25	9.00	7.50	6.43	5.63	5.00	4.09	3.75
4B + memory	15.00	11.25	9.00	7.50	6.43	5.63	5.00	4.09	3.75
4B + reciprocals, square root, and memory	20.00	15.00	12.00	10.00	8.57	7.50	6.67	5.45	5.00
4B + reciprocals, square root, memory, and scientific notation	26.67	20.00	16.00	13.33	11.43	10.00	8.89	7.27	6.67
All above + trigonometric and logarithm functions			28.00	23.33	20.00	17.50	15.56	12.73	11.67

Problem Solution

Problem 43 (PC13-6) presented eight sieve problems, of which the fifth was:

> Take the integers from 3 to N. Circle the 3 and cross off every third remaining number. Circle the next remaining number (4) and cross off every third remaining number. Circle the next remaining number (5) and cross off every third remaining number. List the circled numbers.

The problem called for the 1000th circled number, but this has proved to be difficult. Thomas Hohmann, a student at California State University, Northridge, sifted out the following list:

3	142	12139	1049870
4	212	18208	1574804
5	317	27311	2362205
7	475	40966	3543307
10	712	61448	5314960
14	1067	92171	7972439
20	1600	138256	11958658
29	2399	207383	17937986
43	3598	311074	26906978
64	5396	466610	40360466
95	8093	699914	60540689

Problem 53-P (PC16-15) called for a solution in integers of the equation

$$y^3 - 117x^3 = 5$$

or a proof that no solution exists.

The equation must hold under any modulus. For example, since the right side is odd, then if y^3 is odd, $117x^3$ must also be odd. In particular, the equation must hold modulo 117, for which the x term then is zero and

$$y^3 = 5 \text{ modulo } 117$$

The values of cubes of integers, modulo 117, can be only the following: 0, 1, 8, 18, 26, 27, 44, 53, 64, 73, 90, 91, 99, 109, and 116. The number 5 is not on that list.